Distributed Model Based Event-Triggered Control for Synchronization of Multi-Agent Systems

Davide Liuzza [a] Dimos V. Dimarogonas [a] Mario di Bernardo [b,c] Karl H. Johansson [a]

[a] *ACCESS Linnaeus Center and School of Electrical Engineering, Royal Institute of Technology (KTH), Stockholm, Sweden, {liuzza,dimos,kallej}@kth.se.*

[b] *Department of Engineering Mathematics, University of Bristol, UK, m.dibernardo@bristol.ac.uk.*

[c] *Department of Electrical Engineering and Information Technology, University of Naples Federico II, Italy*

Abstract

This paper investigates the problem of event-triggered control for the synchronization of networks of nonlinear dynamical agents; distributed model-based approaches able to guarantee the synchronization of the overall system are derived. In these control schemes all the agents use a model of their neighborhood in order to generate triggering instants in which the local controller is updated and, if needed, local information based on the adopted control input is broadcasted to neighboring agents. Synchronization of the network is proved and the existence of Zeno behaviour is excluded; an event-triggered strategy able to guarantee the existence of a minimum lower bound between inter-event times for broadcasted information and for control signal updating is proposed, thus allowing applications where both the communication bandwidth and the maximum updating frequency of actuators are critical. This idea is further extended in an asynchronous periodic event-triggered schemes where the agents check a trigger condition via a periodic distributed communication without requiring a model based computation.

Key words: Event-triggered control, synchronization, multi-agent systems.

1 Introduction

The problem of controlling a multi-agent system to reach a cooperative behaviour has been widely exploited in the literature. Specifically, synchronization of dynamical systems has been investigated as a paradigm for more specific behaviours like consensus algorithms and platooning and formation control [Olfati-Saber et al., 2007,Arcak, 2007].
Distributed control algorithms for multi-agent systems have often been realized in continuous time. However, continuous time control laws for such kind of networked systems are not easy or even impossible to implement in real applications where a wireless medium is often exploited to enact the communication.

In order to save the bandwidth and avoid unnecessary updating, the case of event-triggered communication [Tabuada, 2007] among single and double networked integrators has been studied in the recent literature, e.g. [Dimarogonas et al., 2012,Seyboth et al., 2013].
Studies on synchronization of linear systems under an event-triggered framework can be found in [Guinaldo et al., 2011,Liu et al., 2013a] where the control signals are continuous in time and are generated via a model based approach while the communication signals are piecewise constant and based on the error between the real state and the uncoupled model state. Synchronization of linear systems has also been investigated in [Liu et al., 2013b], although the absence of Zeno behaviour [Johansson et al., 1999] is not proved, while in [Persis, 2013] a self-triggered approach is exploited in order to compute the next triggering instant. In this paper we study a novel distributed event-triggered control scheme able to guarantee synchronization of nonlinear multi-agent systems by using distributed information related to each pair of connected agents. The relative information on the state mismatch

⋆ The material in this paper was partially presented at the IFAC meeting NOLCOS 2013. This work was supported by the Swedish Research Council (VR) and the Knut and Alice Wallenberg Foundation. The second author is also affiliated with the KTH Centre for Autonomous Systems. Corresponding author D. Liuzza.

between each pair of connected agents will be considered, in order to generate local events and update the control law. The proposed idea follows a model-based approach, where each agent is equipped with its own embedded processor and it is assumed to know the dynamical model of its neighbours and to predict their state evolutions between any two consecutive triggering events. Both the control and the communication signals will be piecewise constant and, specifically, neighbouring nodes will exchange information about their current (piecewise constant) control input. Such information will allow each node to predict the evolution of its neighbours and evaluate a trigger condition. The proposed scheme solves the problem of achieving synchronization of the interconnected nonlinear systems while guaranteeing a nonzero lower bound for the inter-event time. The existence of such bound is a stronger result than the simple absence of Zeno behaviour, which only excludes accumulation point over a finite time, but does not prevent triggers to get infinitesimally close in time. This advantage allows applications where both the communication bandwidth and the maximum updating frequency of actuators are critical. Furthermore, it also allows the development of an asynchronous periodic event-triggered strategy, where the agents check periodically a trigger condition and decide whether or not update their control input. In this case, no computations based on the model are needed. Such periodic event-triggered scheme represents the other major contribution of this work.
For the sake of brevity, we omit a background section on algebraic graph theory. For more details we refer the reader to [Godsil and Royle, 2001].

2 Model-based Event-triggered Control

Consider N identical dynamical agents of the form:

$$\dot{x}_i = f(t, x_i) + u_i,\ x_i, u_i \in \mathbb{R}^n, t \geq 0,\ \forall i = 1, \dots, N. \tag{1}$$

The aim is to guarantee the emergence of coordinated motion (synchronization) for all the agents by considering a distributed event-triggered control law. More precisely, the average trajectory is defined as

$$\bar{x}(t) = \frac{1}{N} \sum_{j=1}^{N} x_j(t), \tag{2}$$

and the synchronization errors as $e_i(t) = x_i(t) - \bar{x}(t)$, which in stack vector form is $e(t) = \left(e_1^T(t), \dots, e_N^T(t)\right)^T \in \mathbb{R}^{nN}$. We want to achieve either one of the following two objectives:

Bounded synchronization. There exists an arbitrarily small $\epsilon > 0$ such that $\lim_{t\to\infty} \sup ||e(t)||_2 \leq \epsilon$;
Complete synchronization. $\lim_{t\to\infty} ||e(t)||_2 = 0$.

The setup upon which the synchronization analysis will be conducted in Section 3 is now described. Specifically, we assume that each agent is able to exchange information between a subset of the other agents. The resulting communication network, which for the sake of simplicity is assumed to be bidirectional, can be described by an undirected adjacency matrix $A = [a_{ij}]$ defined in the usual way. Furthermore, we assume that each agent is equipped with its own embedded processor able to execute a local control law based on the prediction of the evolution of its neighbours. Thanks to this local information, each node will execute an event-triggered update of its controller. In particular, at each node i we associate:

(1) a time sequence, $\{t_{k^{ij}}\}_{k^{ij}=0}^{\infty} : \mathbb{N} \mapsto [0, +\infty)$, of events corresponding to node i receiving information from node j, where $a_{ij} \neq 0$ and where k^{ij} is the index of the sequence related to the pair (i, j);
(2) a time sequence, $\{t_{k^i}\}_{k^i=0}^{\infty} : \mathbb{N} \mapsto [0, +\infty)$, of instants when node i updates its control input $u_i(t)$, where k^i is the index of the sequence related to the updating of $u_i(t)$.

For any index $k^{ij} \in \mathbb{N}$ (or $k^i \in \mathbb{N}$) we have that $t_{k^{ij}} \leq t_{k^{ij}+1}$ (or $t_{k^i} \leq t_{k^i+1}$).
For each sequence $\{t_{k^{ij}}\}_{k^{ij}=0}^{\infty}$ we introduce the *last function* $l^{ij}(t) : [0, +\infty) \mapsto \mathbb{N}$ defined as $l^{ij}(t) = \arg\min_{k^{ij}\in\mathbb{N}: t\geq t_{k^{ij}}} \{t - t_{k^{ij}}\}$. So, for each time instant t, $t_{l^{ij}(t)}$ is the most recent event occurred to i with respect to j, while with $t_{l^{ij}(t)+1}$ we indicate the next event.
Analogously, we define the function $l^i(t)$ for the sequence $\{t_{k^i}\}_{k^i=0}^{\infty}$.
As will be clear in what follows, the last indices $l^{ij}(t)$ and $l^i(t)$ will be used to generate implicitly the sequences $\{t_{k^{ij}}\}_{k^{ij}=0}^{\infty}$ and $\{t_{k^i}\}_{k^i=0}^{\infty}$.
Note that, although the communication graph is undirected, events related to coupled pairs (i, j) are, in general, not synchronous, so $t_{l^{ij}(t)} \neq t_{l^{ji}(t)}$. For this reason, the sequences $\{t_{k^{ij}}\}_{k^{ij}=0}^{\infty}$ and $\{t_{k^{ji}}\}_{k^{ji}=0}^{\infty}$ are generally different. For the sake of brevity, in what follows we will often omit the explicit dependence of l^{ij} and l^i on time.
The updating law of the sequences $\{t_{k^{ij}}\}_{k^{ij}=0}^{\infty}$ and $\{t_{k^i}\}_{k^i=0}^{\infty}$ will be described in detail in Section 3. Here we anticipate that, for each node i, the control u_i is updated (and so a new event in the sequence $\{t_{k^i}\}_{k^i=0}^{\infty}$ is generated) any time a new event on a connected pair (i, j) happens, i.e., every time there is a new event on one of the sequences $\{t_{k^{ij}}\}_{k^{ij}=0}^{\infty}$, with $j \in \mathcal{N}_i$. So, the latter are subsequences of $\{t_{k^i}\}_{k^i=0}^{\infty}$.

3 Event-triggered Synchronization

In the setup we introduced, each node knows the dynamical model and the value of the initial conditions of its neighbours (or the value of their state at a specific time instant, for example at the first trigger). Therefore, each node i can compute from any event at time $t_{k^{ij}}$ the flow

$\varphi_f(t - t_{k^{ij}}, t_{k^{ij}}, x_j(t_{k^{ij}}))$, $\forall j \in \mathcal{N}_i$. Note that in order to evaluate it, node i must also have information on the current control input $u_j(t)$ acting on each of its neighbours. Later, an algorithm able to guarantee that this information is shared among nodes will be presented. However, we firstly focus on the triggering events occurring at a generic node i.

For all pairs $(i, j) \in \mathcal{E}$ we define the *trigger error*

$$\tilde{e}_{ij}(t) := e_{ij}(t_{l^{ij}}) - e_{ij}(t), \quad t \in [t_{l^{ij}}, t_{l^{ij}+1}), \qquad (3)$$

where $e_{ij}(t) = x_j(t) - x_i(t)$.

The error in (3) is referred to the last and the future trigger instants and is used, as will be clear in what follows, to compute the future trigger instant $t_{l^{ij}+1}$. Similarly $\tilde{e}_{ji}(t)$ is defined for the pair (j, i). Note that, as mentioned earlier, events referred to node i with respect to j are, in general, not synchronous with the events referred to j with respect to i. Indeed, as will be clear in what follows, in general $t_{l^{ij}} \neq t_{l^{ji}}$ since such time instants depend on the whole neighbourhood of node i and j respectively. For this reason, the pair (i, j) is treated here as a directed link and, in general, $\tilde{e}_{ij}(t) \neq \tilde{e}_{ji}(t)$.

For all pairs (i, j), we also define the *trigger function* as $\Xi_{ij}(t, \tilde{e}_{ij}(t)) = \|\tilde{e}_{ij}(t)\|_2 - \varsigma_{ij}(t)$, where $\varsigma_{ij}(t)$ is a continuous-time non-increasing *threshold function* (particular choices of such function will be later considered and analyzed). Then, an event occurs when the following condition is violated

$$\Xi_{ij}(t, \tilde{e}_{ij}(t), \varsigma_{ij}(t)) < 0. \qquad (4)$$

For a generic agent i, the sequences $\{t_{k^{ij}}\}_{k^{ij}=0}^{\infty}$ and $\{t_{k^i}\}_{k^i=0}^{\infty}$ are generated by Algorithm 1 given below, as well as the piecewise constant control input $u_i(i)$, whose value at each update is computed as in (5), with $c > 0$ being a *coupling gain* and $\Gamma = \Gamma^T > 0$ being the *inner coupling matrix.* Such algorithm is run independently at each node of the network. Note that, as every node that triggers changes its control input and broadcasts it to its neighbours (line 10), then all the nodes $j \in \mathcal{N}_i$ can update their dynamic model of i taking into account the new input $u_i(t_{l^i})$ and the current state $x_i(t_{l^i})$ (line 3). So, they will always be able to evaluate the correct value of the flow $\varphi_f(\cdot)$ of node i. Notice also that, since the control input $u_i(t)$ is a piecewise constant function, node i does not need to transmit such information continuously in time, but only when there is a change in its current value.

The initialization of Algorithm 1 happens when at least one node sends the triplet $(t_{0^i}, x_i(t_{0^i}), u_i(t_{0^i}))$ to its neighbours, with t_{0^i} being the time instant when the generic node i broadcasts for the first time its triplet. Then, having received the value of the triplet, all the neighbours can start predicting its evolution and, at the same time, broadcasting their triplets to the transmitting node and to their neighbours. In this way all the nodes of the network can be connected in a finite time. Notice that condition (4) is always verified when a node joins for the first time the network since it computes the first synchronizing control input using the state information coming from its neighbours.

Algorithm 1 Event-triggered update

1: **loop**
2: Integrate the dynamical model $\dot{x}_j = f(t, x_j(t)) + u_j(t_{l^j})$ from the initial condition $x_j(t_{l^j})$ for all the neighbouring nodes $j \in \mathcal{N}_i$, while listening to possible transmission from them;
3: **if** a new value $u_h(t_{l^h})$, with $h \in \mathcal{N}_i$, is received **then** update the dynamical model $\dot{x}_h = f(t, x_h(t)) + u_h(t_{l^h})$;
4: **end if**
5: **if** condition (4) is violated for a node $h \in \mathcal{N}_i$ at a time instant t^* **then**
6: $l^i \leftarrow l^i + 1$ and $t_{l^i} \leftarrow t^*$;
7: $l^{ih} \leftarrow l^{ih} + 1$ and $t_{l^{ih}} \leftarrow t^*$;
8: $e_{ih}(t_{l^{ih}}) \leftarrow e_{ih}(t = t^*)$;
9: $\tilde{e}_{ih} \leftarrow 0$
10: Update the control input to the value

$$u_i(t) = c \sum_{j=1}^{N} a_{ij} \Gamma e_{ij}(t_{l^{ij}}), \qquad t \in [t_{l^i}, t_{l^i+1}), \qquad (5)$$

and broadcast u_i to the neighbourhood $\mathcal{N}_i$;
11: **end if**
12: **end loop**

Remark 2 *If we choose* $\varsigma_{ij}(t) = \varsigma_{ji}(t)$*, Algorithm 1 guaranties that when node i triggers and updates its control, node j also triggers and so we have that* $t_{l^{ij}} = t_{l^{ji}}$*, which in turn implies the symmetry of the coupling strengths between any connected pair* (i, j)*. This fact is a direct consequence of the choice of symmetric threshold functions together with the symmetry of the trigger condition expressed by* (3)–(4)*.*

Note that, when condition (4) in Algorithm 1 is violated for a certain node $h \in \mathcal{N}_i$ (line 5), lines from 6 to 10 could also be replaced by the following fragment of code:

6: $l^i \leftarrow l^i + 1$ and $t_{l^i} \leftarrow t^*$;
7: $l^{ij} \leftarrow l^{ij} + 1$ and $t_{l^{ij}} \leftarrow t^*$, $\forall j \in \mathcal{N}_i$;
8: $e_{ij}(t_{l^{ij}}) \leftarrow e_{ij}(t = t^*)$, $\forall j \in \mathcal{N}_i$;
9: $\tilde{e}_{ij} \leftarrow 0$, $\forall j \in \mathcal{N}_i$;
10: Update the control input to the value

$$u_i(t) = c \sum_{j=1}^{N} a_{ij} \Gamma e_{ij}(t_{l^i}), \qquad t \in [t_{l^i}, t_{l^i+1}). \qquad (6)$$

Basically, in this last case, once the first trigger occurs, say for $\tilde{e}_{ih}(t)$, then not only the current value e_{ih} is updated and the corresponding trigger error (3) reset, but also all other values e_{ij} with $j \in \mathcal{N}_i$. When the above choice is made, we denote the obtained algorithm as Algorithm $1'$.

The control input (5) lead to a diffusively coupled event-triggered dynamical network given by

$$\dot{x}_i(t) = f(t, x_i(t)) + c\sum_{j=1}^{N} a_{ij}\Gamma e_{ij}(t_{l^{ij}}), \tag{7}$$

for $t \in [t_{l^i}, t_{l^i+1})$ and for $i = 1, \ldots, N$. A similar expression is obtained considering the control input (6).

Remark 3 *When using Algorithm 1′, all triggers related to pair (i,j), with $j \in \mathcal{N}_i$, are forced to be synchronous and, moreover, $t_{l^{ij}} = t_{l^{ih}}$ for all $j, h \in \mathcal{N}_i$. Conversely, at a generic time instant t, we have $e_{ij}(t) \neq e_{ji}(t)$. So, as all e_{ij} are updated at the same time, the symmetry is lost of the control actions between coupled pairs (i,j).*

A convergence result for the considered event-triggered control scheme is now given. Before this, let us define

$$\epsilon_\varsigma := \frac{c\sqrt{N}N_{\max}\|\Gamma\|_2}{c\lambda_2(L\otimes\Gamma) - L_f}, \tag{8}$$

and for any constant $\delta > 0$,

$$\alpha := \frac{\delta}{1+\delta}\left[c\lambda_2(L\otimes\Gamma) - L_f\right], \tag{9}$$

where L represents the Laplacian network of the graph and $\lambda_2(L\otimes\Gamma)$ indicates the smallest nonzero eigenvalue of the positive semidefinite matrix $L\otimes\Gamma$.

Theorem 1 *Let us consider the event-triggered connected network (7), where the function $f(t,x)$ is Lipschitz continuous with respect to x with Lipschitz constant L_f and let us choose a coupling gain c such that*

$$L_f - c\lambda_2(L\otimes\Gamma) < 0. \tag{10}$$

Let us consider some constants k_ς such that

$$k_\varsigma \geq \frac{\|e(0)\|_2}{\epsilon_\varsigma}, \tag{11}$$

and λ_ς such that

$$0 < \lambda_\varsigma < \alpha, \tag{12}$$

where ϵ_ς and α are defined in (8) and (9), respectively. We have the following results:

i. If $\lim_{t\to\infty}\varsigma_{ij}(t) = \bar{\varsigma}_{ij}$, with $\bar{\varsigma}_{ij} > 0$ for all i,j such that $a_{ij} \neq 0$, then both Algorithm 1 and Algorithm 1′ guarantee bounded synchronization of the network;
ii. If $\varsigma_{ij}(t) = k_\varsigma e^{-\lambda_\varsigma t}$ for all pairs (i,j) such that $a_{ij} \neq 0$, then Algorithm 1 and Algorithm 1′ guarantee complete synchronization of the network with exponential rate λ_ς.

Furthermore, no Zeno behaviour occurs.

PROOF. The proof is split into two steps. Firstly it is proven that (Step 1) synchronization occurs and then (Step 2) that no Zeno behaviour occurs. Equation (7) can be rewritten as

$$\dot{x}_i = f(t, x_i) + c\sum_{j=1}^{N} a_{ij}\Gamma e_{ij}(t) + \tag{13}$$
$$c\sum_{j=1}^{N} a_{ij}\Gamma\tilde{e}_{ij}(t), \qquad \forall i = 1, \ldots, N.$$

Step 1. Let us consider the candidate Lyapunov function $V(e(t)) = \frac{1}{2}e^T e$ defined in the error space. We obtain

$$\dot{V}(e(t)) = \sum_{i=1}^{N} e_i^T\dot{e}_i = \sum_{i=1}^{N} e_i^T f(t, x_i) - \sum_{i=1}^{N} e_i^T\dot{\bar{x}} -$$
$$ce^T\left(L\otimes\Gamma\right)e + c\sum_{i=1}^{N} e_i^T\sum_{j=1}^{N} a_{ij}\Gamma\tilde{e}_{ij}.$$

Now, from condition (4), taking into account that $\sum_{i=1}^{N} e_i^T\dot{\bar{x}} = 0$, adding and subtracting $\sum_{i=1}^{N} e_i^T f(t,\bar{x})$, where $\sum_{i=1}^{N} e_i^T f(t,\bar{x}) = 0$ since $\sum_{i=1}^{N} e_i^T = 0$, the following inequality holds using the one-sided Lipschitz property [Agarwal and Lakshmikantham, 1993]

$$\dot{V} \leq L_f e^T e - ce^T(L\otimes\Gamma)e + c\|e\|_2\sqrt{N}N_{\max}\|\Gamma\|_2\varsigma(t),$$

where L_f is the Lipschitz constant of the function f and $\varsigma(t) = \max_{i,j}\varsigma_{ij}(t)$. To obtain the previous inequality we have exploited the fact that $\|c\sum_{i=1}^{N} e_i^T\sum_{j=1}^{N} a_{ij}\Gamma\tilde{e}_{ij}\|_2 \leq \|e\|_2\cdot\|\xi\|_2$, with $\xi = \left(c\sum_{j=1}^{N} a_{1j}\Gamma\tilde{e}_{1j}, \ldots, c\sum_{j=1}^{N} a_{Nj}\Gamma\tilde{e}_{Nj}\right)^T$ and that, since $\|\tilde{e}_{ij}\|_2 \leq \varsigma(t)\,\forall i,j$, we can use the bound $\|\xi\|_2 \leq c\sqrt{N}N_{\max}\|\Gamma\|_2\varsigma(t)$.
Writing $e = a\hat{e}$, where $a = \|e\|_2$ is the norm of the error and $\hat{e} = \frac{1}{\|e\|_2}e$ is the unitary vector associated to e and considering that, due to the *Rayleigh-Ritz* theorem [Horn and Johnson, 1987], $\lambda_2(L\otimes\Gamma)e^T e \leq e^T(L\otimes\Gamma)e$, the above inequality can be rewritten as

$$\dot{V}(e) \leq \left(L_f - c\lambda_2(L\otimes\Gamma)\right)a^2 + c\sqrt{N}N_{\max}\|\Gamma\|_2\varsigma(t)a. \tag{14}$$

Now, since c is chosen in order to fulfill inequality (10), then the error trajectory $e(t)$ converges to the invariant region $\|e(t)\|_2 \leq \epsilon$, where

$$\epsilon = \frac{c\sqrt{N}N_{\max}\|\Gamma\|_2\varsigma(t)}{c\lambda_2(L\otimes\Gamma) - L_f}, \tag{15}$$

or, using (8), equivalently $\epsilon = \epsilon_\varsigma\varsigma(t)$. So, if $\lim_{t\to\infty}\varsigma_{ij}(t) = \bar{\varsigma}_{ij}$ is verified, then $\lim_{t\to+\infty}\varsigma(t) = \bar{\varsigma} > 0$ and so bounded synchronization is ensured. Conversely, if $\varsigma_{ij}(t) = k_\varsigma e^{-\lambda_\varsigma t}$ holds, then $\lim_{t\to+\infty} \varsigma(t) = 0$ and so complete synchronization is achieved since the invariant region given by ϵ shrinks with exponential rate λ_ς.

Step 2. We prove next that no Zeno behaviour occurs. The more complicated case (item ii.) of complete synchronization will be firstly analysed, while a simpler reasoning will be later used for the case of bounded synchronization (item i.).

Let us define the strictly decreasing function

$$b(t) = (1+\delta)\epsilon_\varsigma \varsigma(t), \tag{16}$$

where $\delta > 0$ is an arbitrary constant value.

In order to prove that no Zeno behaviour occurs, we first show that for any time instant, inequality

$$\|e(t)\|_2 \le b(t) \qquad \forall t \ge 0, \tag{17}$$

holds. In order to do so, let us note that $\|e(0)\|_2 < b(0)$. Now, since both $e(t)$ and $b(t)$ are continuous, if there is no time instant $\bar{t}$ such that $b(\bar{t}) = \|e(\bar{t})\|_2$, then relation (17) is trivially true. So, let us suppose that such time instant $\bar{t}$ exists. Now, for all $t \ge \bar{t}$ we evaluate the value of $\dot{V}(e)$ when e is such that $\|e\|_2 = b$. More precisely we have that

$$\dot{V}(e)\Big|_{\|e\|_2=b} \le -\delta(1+\delta)\frac{[c\sqrt{N}N_{\max}\|\Gamma\|_2]^2}{c\lambda_2(L\otimes\Gamma) - L_f}\varsigma^2(t).$$

where the above formula has been obtained substituting a with expression (16) in (14). Multiplying and dividing the above relation by $(1+\delta)[c\lambda_2(L\otimes\Gamma) - L_f]$ the following expression is obtained

$$\dot{V}(e)\Big|_{\|e\|_2=b} \le -\alpha b^2, \tag{18}$$

where α has been defined as in (9). Now, since

$$\dot{V}(e)\Big|_{\|e\|_2=b} = \frac{d}{dt}\frac{1}{2}\|e\|_2^2\Big|_{\|e\|_2=b} = b\,\frac{d}{dt}\|e\|_2\Big|_{\|e\|_2=b}, \tag{19}$$

comparing (18) and (19) we get

$$\frac{d}{dt}\|e\|_2\Big|_{\|e\|_2=b} \le -\alpha b. \tag{20}$$

Moreover, considering the decreasing function $B(t) = \frac{1}{2}b^2$ and remembering that $\varsigma(t) = k_\varsigma e^{-\lambda_\varsigma t}$, we have $\dot{B} = -\lambda_\varsigma b^2$. So, using (12) we get $\dot{V}(e)\Big|_{\|e\|_2=b} \le \dot{B} < 0$ or, equivalently

$$\frac{d}{dt}\|e\|_2\Big|_{\|e\|_2=b} \le -\alpha b \le -\lambda_\varsigma b. \tag{21}$$

Since expression (21) holds for all values $b \in [0, b(0)]$, (17) can be obtained by integrating both sides of (21) with respect to time.

We can now show that no Zeno behaviour occurs. Let us consider the dynamics of the error between a generic connected pair of nodes $(i,h) \in \mathcal{E}$. Such dynamics can be expressed as $\dot{e}_{ih}(t) = \dot{x}_h(t) - \dot{x}_i(t)$ thus,

$$\begin{aligned}\dot{e}_{ih} = f(t,x_h) + c\sum_{j=1}^{N} a_{hj}\Gamma e_{hj}(t) + c\sum_{j=1}^{N} a_{hj}\Gamma\tilde{e}_{hj}(t)\\ -f(t,x_i) - c\sum_{j=1}^{N} a_{ij}\Gamma e_{ij}(t) - c\sum_{j=1}^{N} a_{ij}\Gamma\tilde{e}_{ij}(t).\end{aligned}$$

Now, taking the norm of both sides of the above equation into account that f is Lipschitz and that $\|e_{ih}(t)\|_2 = \|x_h(t) - x_i(t) + \bar{x}(t) - \bar{x}(t)\|_2 \le \|e_h(t)\|_2 + \|e_i(t)\|_2 \le 2\|e(t)\|_2$ and recalling relation (17), we obtain

$$\begin{aligned}\|\dot{e}_{ih}(t)\|_2 \le 2\left[L_f + c\|\Gamma\|_2(N_h+N_i)\right]b(t)\\ +c\|\Gamma\|_2(N_h+N_i)\varsigma(t),\end{aligned} \tag{22}$$

where N_i and N_h are the degrees of nodes i and h, respectively, and where we have bounded $\|\tilde{e}_{ij}(t)\|_2$ and $\|\tilde{e}_{hj}(t)\|_2$ with the maximum admissible value of the threshold according to condition (4). Let $q_1 = 2\left[L_f + c\|\Gamma\|_2(N_h+N_i)\right]$ and $q_2 = c\|\Gamma\|_2(N_h+N_i)$. Then, at the last trigger event $t = t_{l^{ih}}$, from (22) we obtain

$$\|\dot{e}_{ih}(t)\|_2 \le q_1(1+\delta)\epsilon_\varsigma k_\varsigma e^{-\lambda_\varsigma t_{l^{ih}}} + q_2 k_\varsigma e^{-\lambda_\varsigma t_{l^{ih}}}, \tag{23}$$

where, we have considered the choice $\varsigma_{ij}(t) = \varsigma(t) = k_\varsigma e^{-\lambda_\varsigma t}$ for all the pairs (i,j) (item ii. in the Theorem statement), with $a_{ij} \ne 0$. Now, in order to prove now that Zeno behaviours do not occur in the network, we show that for all triggering instants $t_{k^{ih}}$ there exists a nonzero lower bound $\tau_m > 0$ such that the next event $t_{k^{ih}+1}$ will satisfy the condition $t_{k^{ih}+1} - t_{k^{ih}} \ge \tau_m$. To do so, let us consider the dynamics of the triggering error $\tilde{e}_{ih}(t)$ at time instants $t > t_l^{ih}$. Clearly, the following considerations will be valid not only for the last event instant $t_{l^{ih}}$ but for all instants $t_{k^{ih}}$, since the sequence $\{t_{k^{ih}}\}_{k^{ih}=0}^{\infty}$ is implicitly defined by the sequence of the last events. It is possible to write

$$\|\tilde{e}_{ih}(t)\|_2 \le \int_{t_{l^{ih}}}^{t} \|\dot{\tilde{e}}_{ih}(s)\|_2\,ds = \int_{t_{l^{ih}}}^{t} \|\dot{e}_{ih}(s)\|_2\,ds.$$

Taking into account inequality (23) and considering $t = t_{l^{ih}} + \tau$ from the above formula, we can write

$$\|\tilde{e}_{ih}(t_{l^{ih}}+\tau)\|_2 \le \left(q_1(1+\delta)\epsilon_\varsigma k_\varsigma e^{-\lambda_\varsigma t_{l^{ih}}} + q_2 k_\varsigma e^{-\lambda_\varsigma t_{l^{ih}}}\right)\tau. \tag{24}$$

Referring to the trigger function (4) with the considered threshold $\varsigma(t_{l^{ih}}+\tau) = k_\varsigma e^{-\lambda_\varsigma(t_{l^{ih}}+\tau)}$, we have that τ_m

solves the equation

$$k_\varsigma e^{-\lambda_\varsigma (t_{l^{ih}}+\tau_m)} = \left(q_1(1+\delta)\epsilon_\varsigma k_\varsigma e^{-\lambda_\varsigma t_{l^{ih}}} + q_2 k_\varsigma e^{-\lambda_\varsigma t_{l^{ih}}}\right)\tau_m.$$

Multiplying both sides of the previous equation by $\frac{1}{k_\varsigma}e^{\lambda_\varsigma t_{l^{ih}}}$ we finally obtain

$$e^{-\lambda_\varsigma \tau_m} = \left(q_1(1+\delta)\epsilon_\varsigma + q_2\right)\tau_m, \tag{25}$$

which implicitly defines τ_m as a non-zero lower bound between any two consecutive triggering instants.

The case of bounded synchronization (item i. of the Theorem statement) is, instead, easier than the case of complete synchronization. Indeed, note that

$$\|e_{ih}(t)\|_2 \leq 2\|e(t)\|_2 \leq 2 \sup_{t' \in [t,+\infty)} \|e(t')\|_2 \leq 2\tilde{b}(t), \tag{26}$$

where $\tilde{b}(t)$ is the nonincreasing piecewise smooth continuous function

$$\tilde{b}(t) = \begin{cases} \|e(t)\|_2 & \text{if } \|e(t)\|_2 > \epsilon_\varsigma \varsigma(t) \\ \epsilon_\varsigma \varsigma(t) & \text{if } \|e(t)\|_2 \leq \epsilon_\varsigma \varsigma(t). \end{cases}$$

So, considering a generic triggering event at $t = t_{l^{ih}}$, inequality (22) can be bounded as

$$\|\dot{e}_{ih}(t)\|_2 \leq q_1 \tilde{b}(t_{l^{ih}}) + q_2 \varsigma(t_{l^{ih}}), \quad \forall t \geq t_{l^{ih}}, \tag{27}$$

where the same positions of q_1 and q_2 as done in equation (23) have been used in order to simplify the notation. Integrating both sides of (27) with respect to time, a nonzero lower bound $\tau_{ih}(t_{l^{ih}})$ for the inter-event time between the last trigger event t_l^{ih} and the next one $t_{l^{ih}+1}$ for the generic pair (i,h) is

$$\tau_{ih}(t_{l^{ih}}) = \frac{\bar{\varsigma}_{ih}}{q_1 \tilde{b}(t_{l^{ih}}) + q_2 \varsigma(t_{l^{ih}})}. \tag{28}$$

This completes the proof.

Remark 4 *Notice that choosing a high value of λ_ς allows to speed up the convergence rate. However,* (25) *shows that a faster synchronization reduces the value of the inter-event bound and so increases the frequency of the triggers.*

Remark 5 *Note that Theorem 1 holds for both Algorithm 1 and Algorithm 1′ since the proof is independent on the updating criterion of e_{ij}. Since in Algorithm 1′ all e_{ij} with $j \in \mathcal{N}_i$ are updated at the same time instant t_{l^i} and the corresponding errors $\tilde{e}_{ij}$ are reset, both for bounded and complete synchronization there implicitly exists a non-zero lower bound between any two consecutive updating events of the control law. For this reason, Algorithm 1′ can be implemented in all applications where constraints on actuators do not allow to change the control input arbitrarily fast.*

4 Periodic event-triggered synchronization

The scheme presented in the previous section can be easily modified in order to derive a periodic event-triggered synchronization setup, where the agents communicates in an asynchronous way and with clocks of possibly different periods. Such scheme exploits the advantages of having a nonzero lower bound for the inter-event times in Algorithm 1′, which turns to be useful for a periodic event detection of a trigger condition. Specifically, we no longer require a model based approach and we consider here for each agent i a periodic checking sequence $\{T_{p^i}\}_{p^i=0}^{\infty}$, where the agent obtains its neighbours' state value. Such sequence is computed considering the sampled times $T_{p^i} = t_{0^i} + p^i \tau_i$, with $\tau_i \leq \tau_m$, where τ_m is the solution of (25). For the sake of brevity and without loss of generality, we present here the case of a constant period τ_i, but the same setup and the same analysis can be adopted in the case of a time-varying $\tau_i(T_{p^i}) \leq \tau_m$, thus resulting in a cyclic event-triggered detection, instead of a periodic one. More specifically, at each T_{p^i}, agent i obtains the measurement of its own state and the one of its neighbours and evaluates, for all $h \in \mathcal{N}_i$, the value $\tilde{e}_{ih}(T_{p^i})$ given in (3), deciding whether or not to trigger an update of its piecewise constant control input (6) according to a suitable trigger condition. So, the resulting updating sequence $\{t_{k^i}\}_{k^i=0}^{\infty}$ is a subsequence of the periodic checking sequence, namely $\{t_{k^i}\}_{k^i=0}^{\infty} \subseteq \{T_{p^i}\}_{p^i=0}^{\infty}$. Considering the same constants given in Section 3, the periodic event-triggered synchronization algorithm is reported below as Algorithm 2.

Algorithm 2 Periodic event-triggered update

- **for** $p^i = 0, 1, \ldots$ **do**
  - At each instant $T_{p^i} = t_{0^i} + p^i \tau_i$, obtain neighbours' state information and evaluate $\tilde{e}_{ih}(T_{p^i})$, with $h \in \mathcal{N}_i$.
  - Compute $\bar{\tau}_i(T_{p^i}) = \min_{h \in \mathcal{N}_i} \tau_{ih}^*(T_{p^i})$, with $\tau_{ih}^*(T_{p^i})$ the solution of the implicit equation

$$e^{-\lambda_\varsigma \tau_{ih}^*(T_{p^i})} = \frac{1}{k_\varsigma}\|\tilde{e}_{ih}(T_{p^i})\|_2 e^{\lambda_\varsigma T_{p^i}} + (q_1(1+\delta)\epsilon_\varsigma + q_2)\tau_{ih}^*(T_{p^i}) \tag{29}$$

  - **if** $\bar{\tau}_i(T_{p^i}) < \tau_i$ **then**
    - $l^i \leftarrow l^i + 1$ and $t_{l^i} \leftarrow T_{p^i}$;
    - $\tilde{e}_{ih} \leftarrow 0$;
    - Update the control input $u_i(t_{l^i})$ to the value in (6);
  - **end if**
- **end for**

For the periodic event-triggered scheme we can give the following result.

Theorem 2 *Let us consider network* (7)*, where the function $f(t,x)$ is Lipschitz continuous with respect to* *x with Lipschitz constant L_f and let us choose a coupling gain c and a function $\varsigma(t) = k_\varsigma e^{-\lambda_\varsigma t}$ such that the inequalities (10)-(12) of Theorem 1 hold. Then, the periodic event-triggered control scheme given in Algorithm 2 guarantees complete synchronization of the network with exponential rate λ_ς.*

PROOF. The proof can be obtained following similar steps of those in the proof of Theorem 1. In particular, the key point is to prove that the trigger condition $\bar{\tau}_i(T_{p^i}) < \tau_i$ guarantees $\|\tilde{e}_{ih}(t)\|_2 \leq \varsigma(t)$ for all $t \geq 0$ and for all the connected pairs (i, h). To do so, let us consider an induction argument starting from a generic update time $t_{l^i} = T_{l^i}$. Following the same steps leading to (22), we obtain

$$\|\dot{e}_{ih}(t)\|_2 \leq q_1(1+\delta)\epsilon_\varsigma k_\varsigma e^{-\lambda_\varsigma T_{l^i}} + q_2 k_\varsigma e^{-\lambda_\varsigma T_{l^i}}, \quad (30)$$

which is the same inequality of (23) where we have substituted $t_{l^{ih}}$ with T_{l^i}. Now, considering that $\tilde{e}_{ih}(t) = \tilde{e}_{ih}(T_{l^i}) + \int_{T_{l^i}}^{t} \dot{\tilde{e}}_{ih}(s)ds$, analogously to what was done for Theorem 1, evaluating the norm of both sides and substituting (30), we obtain $\|\tilde{e}_{ih}(T_{l^i} + \tau_{ih})\|_2 \leq \|\tilde{e}_{ih}(T_{l^i})\|_2 + \left(q_1(1+\delta)\epsilon_\varsigma k_\varsigma e^{-\lambda_\varsigma T_{l^i}} + q_2 k_\varsigma e^{-\lambda_\varsigma T_{l^i}}\right)\tau_{ih}$. Following similar steps to those taken to obtain equation (25), we get the implicit equation (29) in the variable $\tau^*_{ih}(T_{p^i})$. Since at $t_{l^i} = T_{l^i}$ we have $\|\tilde{e}_{ih}(t)\|_2 = 0$ due to line 6 in Algorithm 2, equations (25) and (29) are identical, providing the same solution τ_m. So, at the following checking instant $\|\tilde{e}_{ih}(T_{p^i+1})\|_2 \leq \varsigma(T_{p^i+1})$. Solving again equation (29) at T_{p^i+1}, if $\bar{\tau}_i(T_{p^i+1}) \geq \tau_i$ then at the checking instant T_{p^i+2} we will satisfy again $\|\tilde{e}_{ih}(T_{p^i+2})\|_2 \leq \varsigma(T_{p^i+2})$, otherwise a trigger is generated and the control input is updated. By induction, the reasoning can be iterated for all the future instants,while the stability proof follows exactly the same steps as in Step 1 of Theorem 1.

5 Numerical Examples

We consider a network of identical Chua circuits, a paradigmatic nonlinear example of chaotic behaviours which has been considered as a testbed for both numerical and experimental analysis of theoretical synchronization strategies and for applications to communication [Gámez-Guzmán et al., 2009,de Magistris et al., 2012]. The dynamical model of a single system is $\dot{x}_i = f(x_i)$ given by $\dot{x}_{i1} = \alpha\left[x_{i2} - x_{i1} - \varphi(x_{i1})\right]$; $\dot{x}_{i2} = x_{i1} - x_{i2} + x_{i3}$; $\dot{x}_{i3} = -\beta x_{i2}$, with $\alpha = 10$, $\beta = 17.30$, and $\varphi(x_{i1}) = bx_{i1} + (a-b)(|x_{i1}+1| - |x_{i1}-1|)/2$, with $a = -1.34$, $b = -0.73$. For this vector field it is possible to evaluate an upper bound for the Lipschitz constant $L_f = 34.2$; a network of five Chua circuits over a connected random graph is simulated and the matrix Γ is the identity matrix for the sake of simplicity; a value of the minimum coupling guaranteeing inequality (10) is $c = 13.7$. Simulations have been performed applying Algorithm 1 and setting an identical static threshold $\varsigma_{ij}(t) = \bar{\varsigma}$ for all connected pairs (i, j), with $\bar{\varsigma} = 0.1$. The synchronization of the chaotic trajectories is obtained within the first 2 s (figures are omitted here for the sake of brevity). Also, simulations have been carried out for Algorithm 1′ considering identical exponential threshold functions $\varsigma_{ij} = k_\varsigma e^{-\lambda_\varsigma t}$ with $k_\varsigma = 1$ and $\lambda_\varsigma = 0.5$ (again figures are omitted for the sake of brevity). In this case, the exponential synchronization of the network is obtained within 5 s. The number of triggers for each node in time intervals of unitary length for the first 5 s of simulation is reported in Tab. 1 and Tab. 2 for the case of static threshold with Algorithm 1 and for the case of exponential threshold with Algorithm 1′, respectively. Observe how the first approach generates an higher number of triggers than the second one. Simulations have also been carried out for the case of the same static threshold with Algorithm 1′, showing better performance than the case with Algorithm 1.

	$[0,1)$ s	$[1,2)$ s	$[2,3)$ s	$[3,4)$ s	$[4,5)$
node 1	32	33	26	25	22
node 2	35	25	27	39	33
node 3	29	25	19	22	23
node 4	32	24	22	27	36
node 5	15	15	14	21	23

Table 1
Number of triggers in unitary intervals for the network of Chua systems: static thresholds

Simulations for the case of periodic event-triggered synchronization developed in Algorithm 2 have been conducted, considering for (29) the same parameters of the case of identical exponential threshold. This led to a $\tau_m = 0.17$ ms and so, $\tau_i \leq \tau_m$ have been randomly assigned accordingly for each node. As illustrated in Tab. 3, a significant higher number of triggers is generated in the periodic case due to the conservativeness of the approach, that does not rely on model-based computations. Although, a faster convergence is obtained (within 0.5 s) due to the reduced error mismatch on the connected pairs of agents.
Finally, for the sake of comparison, a time-triggered control protocol where all the nodes update their control law following a centralized sampling of period $T_s = 60$ ms has been carried out. Such sampling period corresponds to the average of all the inter-event intervals obtained for the case of exponential thresholds and leads the network to instability.

6 Conclusions

A model-based approach where connected agents broadcast input information has been considered and results

	$[0,1)$ s	$[1,2)$ s	$[2,3)$ s	$[3,4)$ s	$[4,5)$
node 1	16	14	17	16	17
node 2	19	25	26	26	22
node 3	7	12	18	13	13
node 4	0	0	0	0	20
node 5	0	0	0	0	10

Table 2
Number of triggers in unitary intervals for the network of Chua systems: exponential thresholds

	$[0,1)$ s	$[1,2)$ s	$[2,3)$ s	$[3,4)$ s	$[4,5)$
node 1	5882	5883	5882	5882	5883
node 2	6667	6666	6667	6667	6666
node 3	1000	1000	1000	1000	1000
node 4	8333	8333	8334	8333	8333
node 5	7143	7142	7143	7143	7143

Table 3
Number of triggers in unitary intervals for the network of Chua systems: periodic event-triggered

have been given for bounded synchronization and for exponential synchronization. The absence of Zeno behaviour has been proven guaranteeing a lower bound for the inter-event times between consecutive updates. This fact allowed to extend the results to an asynchronous periodic event-triggered setup, where the agents periodically gather neigbours' state information and check for a trigger condition in order to decide whether or not update their control input. In this latter scheme, a model-based information is no longer required.
The proposed strategies have been shown to be promising for synchronization of nonlinear systems.

References

[Agarwal and Lakshmikantham, 1993] Agarwal, R. P. and Lakshmikantham, V. (1993). *Uniqueness and Nonuniqueness Criteria for Ordinary Differential Equations*. Singapore: World Scentific.

[Arcak, 2007] Arcak, M. (2007). Passivity as a design tool for group coordination. *IEEE Trans. on Automatic Control*, 52(8):1380–1390.

[de Magistris et al., 2012] de Magistris, M., di Bernardo, M., Tucci, E. D., and Manfredi, S. (2012). Synchronization of networks of non-identical Chua circuits: analysis and experiments. *IEEE Transactions on Circuits and Systems*, 59:1029–1041.

[Dimarogonas et al., 2012] Dimarogonas, D. V., Frazzoli, E., and Johansson, K. H. (2012). Distributed event-triggered control for multi-agent systems. *IEEE Transactions of Automatic Control*, 57:1291–1297.

[Gámez-Guzmán et al., 2009] Gámez-Guzmán, L., Cruz-Hernández, C., López-Gutiérrez, R. M., and García-Guerrero, E. E. (2009). Synchronization of Chua's circuits with multi-scroll attractors: Application to communication. *Communications in Nonlinear Science and Numerical Simulation*, 14:2765–2775.

[Godsil and Royle, 2001] Godsil, C. D. and Royle, G. (2001). *Algebraic Graph Theory*. Springer.

[Guinaldo et al., 2011] Guinaldo, M., Dimarogonas, D. V., Johansson, K. H., Sánchez, J., and Dormido, S. (2011). Distributed event-based control for interconnected linear systems. In *50th IEEE Conference on Decision and Control and European Control Conference*.

[Horn and Johnson, 1987] Horn, R. A. and Johnson, C. R. (1987). *Matrix Analisis*. Cambridge University Press.

[Johansson et al., 1999] Johansson, K. H., Egerstedt, M., Lygeros, J., and Sastry, S. S. (1999). On the regularization of Zeno hybrid automata. *Systems and Control Letters*, 38:141–150.

[Liu et al., 2013a] Liu, T., Cao, M., Persis, C. D., and Hendrickx, J. (2013a). Distributed event-triggered control for synchronization of dynamical networks with estimators. In *Proceeding of IFAC NecSys 2013*.

[Liu et al., 2013b] Liu, T., Hill, D. J., and Liu, B. (2013b). Synchronization of dynamical networks with distributed event-based communication. In *51st IEEE Conference on Decision and Control*, pages 7199–7204.

[Olfati-Saber et al., 2007] Olfati-Saber, R., A.Fax, J., and Murray, R. (2007). Consensus and cooperation in networked multi-agent system. In *Proceedings of the IEEE*, volume 95, pages 215–233.

[Persis, 2013] Persis, C. D. (2013). On self-triggered synchronization of linear systems. In *4th IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys)*.

[Seyboth et al., 2013] Seyboth, G. S., Dimarogonas, D. V., and Johansson, K. H. (2013). Event-based broadcasting for multi-agent average consensus. *Automatica*, 49:245–252.

[Tabuada, 2007] Tabuada, P. (2007). Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Tansactions on Automatic Control*, 52:1680–1684.